

TD3 Asm Cortex-M3

3 IMACS 2010-2011

Vincent MAHOUT

Abstract— Le but de ce TD est de passer en revue, c'est à dire proposer des codages d'algorithme, les différentes structures algorithmiques basiques.

I. SI...SINON

Soit l'algorithme suivant :

```

Nombre = Saisie()
Compte = Compte +1
Si (Nombre < 0)
    Nombre ← -Nombre
FinSi
Si (Nombre < (Max-Biais))
    Nombre ← Nombre+Biais
Sinon
    Nombre ← Max
FinSi
Table[Compte] = Nombre

```

Commentaire : Ne pas oublier les déclaration de sections....

Question 1: *Compte* et *Max* sont deux 1/2 mots (16 bits) et *Biais* est un octet. Écrivez les lignes qui permettent de faire la réservation de ces variables. On donnera des valeurs initiales (laissées à votre choix) à *Max* et *Biais*. *Compte* sera initialisé à -1. *Table* sera un tableau de *Taille* demi mots, *Taille* étant une constante déclarée en début de listing.

Commentaire : Rien de compliqué - Attention aux accès signés/non signés et octet/demi mot. Difficultés à prévoir sur le fait qu'il faille calculer (Max-Biais) avant de faire le test. J'insiste en cours sur le codage correct d'un SI à savoir si l'on a SI(COND) on code un Branch non(COND) pour éviter le traitement. Cela reste vrai même pour le SI..SINON pour respecter l'ordre de l'algo : le code de TraitSI doit être AVANT le code de TraitSinon

Question 2: Proposez un codage de cet algorithme en respectant les consignes suivantes :

- *R0* servira de registre temporaire pour contenir les adresses des entités mémoires auxquelles il sera nécessaire d'accéder.
- *R1* servira de pointeur sur *Table*
- *Compte* sera lu en mémoire puis réécrit en mémoire après incrémentation.

II. EXPORT EN PROCÉDURE ET BOUCLE TANT QUE

On se propose d'exporter le formatage de la données saisie dans une procédure. L'algorithme précédent devient alors :

```

Nombre = Saisie()
Compte = Compte +1
Nombre = Formatage(Nombre)
Table[Compte] = Nombre

```

où *Formatage* est la procédure qui contient l'ensemble des structures Si...Sinon.

Commentaire : Poser des jalons sur le passage d'arguments à une procédure

Question 3: Écrivez la procédure en supposant qu'elle connaît les variables globales *Max* et *Biais*. On supposera que l'argument *Nombre* est passé en entrée de cette procédure avec le registre *R4* et que c'est par ce même registre qu'elle retournera son résultat.

Commentaire : Insistez pour que le test se fasse bien au début de la procédure Tantque...Faire...FinTantque - Ce qui nécessite le saut non conditionné à la fin

Question 4: Insérez à cet algorithme une structure *TantQue* pour que le processus de saisie/formatage/stockage soit effectué tant que la valeur 0 n'a pas été saisie.

Commentaire : Test de condition avec une composition en ET .. là aussi gare aux sauts imbriqués dans tous les sens

Question 5: Comment doit-on modifier le test de ce *TantQue* pour être sûr de ne pas dépasser les capacités de stockage dans *Table*? Codez cette modification.

III. BOUCLE POUR

Commentaire : Personnellement je suis pour ! Le passage d'arguments est laissé libre mais doit être réfléchi a minima

Question 6: Écrivez une procédure (ainsi que son appel) qui permettent de calculer la moyenne de *Table* précédemment saisie. Cette procédure s'appuiera sur une structure de boucle *Pour* utilisant la variable *Compte*.